UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/943,964	08/31/2001	Shawn S. Cornelius	10022/55	1551

28164 7590 09/21/2006

ACCENTURE CHICAGO 28164
BRINKS HOFER GILSON & LIONE
P O BOX 10395
CHICAGO, IL 60610

EXAMINER	
DIVECHA, KAMAL B	

ART UNIT	PAPER NUMBER
2151	

DATE MAILED: 09/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/943,964
Filing Date: August 31, 2001
Appellant(s): CORNELIUS ET AL.

John C. Freeman
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed June 20, 2006 appealing from the Office action mailed March 20, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest in contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Ahmed, U. S. Patent No. 6,813,634 B1, issued on Nov. 2, 2004, but filed on Feb. 3, 2000.

Sato et al., U. S. Patent No. 6,718,482 B2, issued on Apr. 6, 2004, but filed on Jan. 19, 2001.

Ullman, Pub. No., US 2002/0112039 A1, issued on Aug. 15, 2002, but file d on Dec. 15, 2000.

Short et al., U. S. Patent No. 6,178,529 B1, issued on Jan. 23, 2001, but filed on Nov. 3, 1997.

Pocrass, U. S. Patent No. 5,428,806, issued on Jun. 27, 1995, but filed on Jan. 22, 1993.

Hirosawa et al., U. S. Patent No. 5,237,677, issued on Aug. 17, 1993, but filed on Nov. 6, 1990.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Specification

The specification is objected to under 35 U.S.C. § 112, first paragraph, as failing to adequately teach how to make and use the invention, i.e., failing to provide an enabling disclosure.

The test to be applied under the written description portion of 35 U.S.C. § 112, first paragraph, is whether the disclosure of the application as originally filed reasonably conveys to the artisan that the inventor had possession at that time of later claimed subject matter. Vas-Cat, Inc. v. Mahurkar, 935 F. 2d 1555, 1565, 19 USPQ2d 111, 1118 (Fed. Cir. 1991), reh'rg denied (Fed. Cir. July 8, 1991) and reh'rg, en banc, denied (Fed. Cir. July 29, 1991).

The applicants have failed to provide an enabling disclosure in the detailed description of the embodiment. The specification is objected to under 35 U.S.C. § 112, first paragraph, as failing to support the subject matter set forth in these claims.

The claims recite the limitation of "a fault detector associated with the first software stage component and the second software stage component to detect a fault in the remote software module by detecting whether the data message or a derivative thereof flows **entirely** through at least one of the first stage software component and the second stage software component", "...hence, flow **entirely** through at least one of the first stage software component and the second stage software component".

However, the specification merely suggests that the process of identifying a deficient software component of the installed remote software module as any of said software stage

components that **block or disrupts** the flow of data message between two adjacent nodes (figure 8).

There is no indication whatsoever in the specification of the fact that the process of detecting fault in the software module is conducted by detecting whether the data message or a derivative thereof flows **entirely** through at least one of the software stages.

Claim Rejections - 35 USC § 112

Claims 1-21 are rejected under 35 U. S. C. 112, first paragraph, for the same reasons as set forth in the objection to the specification.

Claim Rejections - 35 USC § 102

Claim 22 is rejected under 35 U.S.C. 102(e) as being anticipated by Ahmed (U. S. Patent No. 6,813,634 B1).

As per claim 22, Ahmed discloses a method of monitoring a business-to-business system, the method comprising: transmitting a status code from a base data processing system to a remote data processing system via a communications network (fig. 2 item #21); receiving the status code at a data receiver in the remote data processing system (fig. 2 item #21); inputting the status code into a remote software module of the remote data processing system (networked PC in figure 2 inherently inputs the status code or ping into a remote software module); determining whether the remote software module provides a logical data path of continuity to the status code; outputting the status code from an output of the remote software module if the determining determines that the remote software module provides a logical data path of continuity to the status code (fig. 2 item #22: the step of determining is inherent to the step of outputting, i.e. Ahmed's system inherently determines that the remote software module provides a logical data

path of continuity to the status code because the networked PC outputs the status code because the software module provides logical path of continuity to the status code); and transmitting the outputted status code back to the base data processing system via the communications network as a feedback indicative of the proper end-to-end continuity of communications in a business-to-business environment (fig. 2 item #22, 23, fig. 3 item #31 and col. 2 L29-65: note that if a response or reply is received from networked PC, than there must be an end-to-end continuity of communications).

Claim Rejections - 35 USC § 103

Claims 1-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ahmed (U. S. Patent No. 6,813,634 B1) in view Sato et al., (hereinafter Sato, U. S. Patent No. 6,718,482 B2).

As per claim 1, Ahmed discloses a remote data processing system comprising: a data receiver for receiving the data message (fig. 1 and fig. 2); a remote software module arranged to receive the data message from the data receiver (fig. 1: PC are inherently associated with the software modules such as operating system that usually includes plurality of software stages cascaded with each other); and a fault detector associated with the remote software module to detect a fault in the remote software module by detecting whether the data message or a derivative thereof flows entirely through the remote software module (fig. 1 item #2, 4, 5: the computers associated with the network 2, 4, 5 are inherently equipped with the remote software module or an operating system) by detecting whether the data message or a derivative thereof flows entirely through at least one of the first stage software component and the second stage software component (fig. 2 item #21, 22, 20, 23 and col. 2 L29-64: software module receives the

ping message and replies to the ping message if the software module in PC is functioning, in other words the ping message flows through the software module in PC and the reply is sent, please note that if the software module was not functioning than the software module would not be able to process the ping message, hence no reply would be sent in this case: all interpreted as whether the data message flows through the software module), however Ahmed does not disclose that the remote software module includes a first stage software component cascaded with a second stage software component (please note that Ahmed might inherently disclose the remote software module including first stage software module cascaded with the second stage).

Sato, from the same field of endeavor discloses a remote software module arranged to receive the data message from the data receiver, the remote software module including at least a first stage software component cascaded with a second stage software component (fig. 1 item #112, 105 and item # 110).

Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ahmed in view of Sato in order to include a remote software module including at least first stage software component cascaded with a second stage software component, since Sato teaches cascading the first software stage with the second software stage component.

One of ordinary skilled in the art would have been motivated because cascading the plurality of software modules in a single computer would have enabled the communications to the different computers on the network and would have enabled the process of monitoring the software faults in the first software environment (Sato, col. 2 L47-67).

As per claim 2, Ahmed discloses the system comprising a remote status reporter for reporting a status message on at least one of the remote software module and hardware of the remote data processing system (col. 2 L28-45: a reply is sent in response to ping message, since the PC is capable of replying to the ping query message, it must be equipped with the status reporter for reporting the status).

As per claim 3, Ahmed discloses the system comprising a database for storing detected faults, stage identifiers, and fault descriptions outputted by the fault detector (fig. 1 item #13, 11 and fig. 5 item #13, 11, col. 5 L50-67).

As per claim 4, Ahmed discloses the system comprising a database for storing the status data on corresponding components of a remote data processing system (fig. 1 item 11, 13, fig. 5 item #11, 13 and col. 5 L50-57).

As per claim 5, Ahmed discloses the system comprising a database, the fault detector logging one or more error messages into the database (col. 5 L20-37).

As per claim 6, Ahmed discloses the system comprising feedback generator associated with the remote software module, the feedback generator receiving a status code outputted from the remote software module and forwarding the status code to a transmitter for transmission via a communications network (fig. 2 item #22 and item #23: since the PC and the associated software module provides a feedback or a reply, there must be a feedback generator associated with the PC or the software module that is able to output a status code from the remote software module and forwarding the status code to transmitter for transmission via communications network).

As per claim 7, Ahmed discloses the feedback generator associated with the remote software module, the feedback generator generating a status code for a transmitter upon detection

of a status code from the remote software module (fig. 2 item #22 and item #23: since a reply of Ok is sent by the PC and/or remote software module, there must be a feedback generator that generates reply or status code upon the detection from the software module).

As per claim 8, Ahmed does not disclose the system wherein the fault detector has logical connections including a connection with an input of the first software stage component, an output of the first software stage component, and an output of the second software stage component. Sato discloses the fault detector having logical connections including a connection with an input of the first software stage component, an output of the first software stage component, and an output of second software stage component (fig. 1 item #101). Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ahmed in view of Sato, wherein fault detector would have logical connections with an input and output of the first stage software component and an output of the second software stage component. One of ordinary skilled in the art would have been motivated because it would have identified the occurrence and the location of the faults.

As per claim 9, Ahmed discloses the system comprising a fault detector that identifies the first software stage as a faulty software component if the data message is present at an input of the first software stage, but not the output of the first software stage (fig. 2 and col. 2 L29-65).

As per claim 10, Ahmed does not disclose the system wherein the fault detector identifies the second software stage as a faulty software component if the data message is present at an input of the second software stage, but not the output of the second software stage. But, it would have been obvious to the one of ordinary skilled in the art at the time the invention was made to modify Ahmed in order to detect fault in the second software stage by pinging the second

software stage component and monitoring the response or reply, since Ahmed teaches the process of pinging and monitoring the responses in the remote software module (fig. 2 and col. 2 L29-65). One of ordinary skilled in the art would have been motivated because it would have detected faults in the software module (Ahmed, col. 2 L29-65).

As per claim 11, Ahmed discloses the system comprising a fault detector that identifies the first software stage as a faulty software component if a derivative of the data message (please note derivative of a data message is simply interpreted as a data message or a query message) is present at an input of the first software stage, but not the output of the first software stage (fig. 2 and col. 2 L29-65).

As per claim 12, Ahmed does not disclose the system wherein the fault detector identifies the second software stage as a faulty software component if a derivative of the data message is present at an input of the second software stage, but not the output of the second software stage. But, it would have been obvious to the one of ordinary skilled in the art at the time the invention was made to modify Ahmed in order to detect fault in the second software stage by pinging the second software stage component and monitoring the response or reply, since Ahmed teaches the process of pinging and monitoring the responses in the remote software module (fig. 2 and col. 2 L29-65). One of ordinary skilled in the art would have been motivated because it would have detected faults in the software module whether it's a first stage software component or the second stage software component (Ahmed, col. 2 L29-65).

Claims 13, 16, 18 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ullman (Pub. No.: US 2002/0112039 A1) in view Sato et al., (hereinafter Sato, U. S. Patent No. 6,718,482 B2).

As per claim 13, Ullman discloses a method for monitoring a remote data processing system, the method comprising: having a remote data processing system receive a data message from a base data processing system via a communications network that is external to the remote data processing system (fig. 2A item #214, 218; fig. 5A; pg. 5 [0069]; pg. 6 [0084]); cascading at least a first stage software component to form an installed software module of the remote data processing system for accepting the received data message (fig. 2E item #230 and item #236; pg. 5 [0073]: please note that an operating system associated with the endpoint device is an remote software module that inherently includes at least a first stage software component and second stage software component); identifying a deficient software component of the installed remote software module as any of said software stage components that blocks or disrupts the flow of the data message between the two adjacent logical nodes (fig. 9F item #966, 968, 970, 972, 976), however Ullman does not disclose the process of detecting the data message or a derivative at a group of logical nodes within the installed remote software module to determine flow of the data message, or a derivative thereof, between the logical nodes and, hence, flow entirely through at least one of the first stage software component and the second stage software component.

Sato, from the same field of endeavor discloses the process of detecting the data message or a derivative at a group of logical nodes within the installed remote software module to determine the flow of data message, or a derivative thereof, between the logical nodes and,

hence, flow entirely through at least one of the first stage software component and the second stage software component (col. 8 L28-52 and fig. 8).

Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ullman in view of Sato, in order to detect the data message or a derivative at a group of logical nodes within the installed remote software module to determine flow of the data message.

One of ordinary skilled in the art would have been motivated because it would monitored the software faults within the monitored computer (Sato, see abstract, col. 8 L28-52).

As per claim 16, Ullman discloses the process of routing the status code from the base data processing system via the communications network to the remote data processing system (fig. 9F item #966) and routing the status code from the remote data processing system to the base data processing system (fig. 9F item #970, item #972) to indicate that the continuity of at least one logic path traversed by the status code.

As per claim 18, Ullman discloses the process of determining that one of the at least a first stage software component and a second stage software component immediately following the last detected data message is at fault (fig. 9F item #966, 968, 970, 976).

As per claim 21, Ullman discloses the process of archiving a fault analysis report in a database associated with the remote data processing system (pg. 10 [0144] and pg. 14 [0187-0188]).

Claims 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ullman (Pub. No.: US 2002/0112039 A1) in view Sato et al., (hereinafter Sato, U. S. Patent No. 6,718,482 B2), and further in view Short et al., (hereinafter Short, U. S. Patent No. 6,178,529 B1).

As per claim 14, Ullman in view of Sato does not disclose the process of passing a status code, distinct from the data message, from at least input of the installed remote software module to an output of the installed remote software module to indicate that the installed remote software module is operational.

Short, from the same field of endeavor discloses the process of sending the periodic messages, called heartbeats, to other resources such as physical device or an application to provide a mechanism for detecting that the communication path is good and the other systems such as physical device or an application are operational (col. 5 L1-10).

Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ullman and Sato in view of Short, in order to pass the status code distinct from the status message, from an input of the installed remote software module to an output of the installed software module to indicate that the installed remote software module is operational.

One of ordinary skilled in the art would have been motivated because it would have provided a mechanism for detecting that the software module is operational (Short, col. 5 L1-5).

As per claim 15, Ullman in view of Sato does not disclose the process of passing a status code from at least an input of the communications network to an output of the communications network to indicate that the communications network is operational. Short from the same field of

endeavor discloses the process of sending periodic messages to counterpart components on the other systems to provide a mechanism for detecting that the communication path or network is operational or good (col. 5 L1-5). Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ullman and Sato in view of Short, in order to pass a status code from an input of the communications network to an output of the communications network to indicate that the communication network is operational. One of ordinary skilled in the art would have been motivated because it would have provided a mechanism for detecting that the communication path or network is good or operational (Short, col. 5 L1-5).

Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ullman (Pub. No.: US 2002/0112039 A1) in view Sato et al., (hereinafter Sato, U. S. Patent No. 6,718,482 B2), and further in view Pocrass (U. S. Patent No. 5,428,806).

As per claim 17, Ullman in view of Sato does not explicitly disclose the process of tapping into a logical data path between the first stage software component and the second stage software component to detect whether each of the first and second stage software components are functioning.

Pocrass explicitly discloses the process of tapping into the logical data path extending between two modules (col. 18 L54-59).

Therefore, it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to incorporate the teaching of Pocrass as stated above with Ullman in view of Sato, in order to tap into the data path to detect whether the software modules or components are functioning.

One of ordinary skilled in the art would have been motivated because it would have provided an easy way to check the status of the data path, configure, operate and would have maintained the data paths (Pocrass, col. 22 L34-41, col. 3 L26-31).

Claims 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ullman (Pub. No.: US 2002/0112039 A1) in view Sato et al., (hereinafter Sato, U. S. Patent No. 6,718,482 B2), and further in view of Hirosawa et al., (hereinafter Hirosawa, U. S. Patent No. 5,237,677).

As per claim 19, Ullman in view of Sato does not disclose the process of assigning stage identifiers to distinguish the at least a first stage software component and a second stage software component from one another and to identify a faulty stage.

Hirosawa, from the same field of endeavor discloses the process of assigning stage identifiers to distinguish the at least a first stage software component and a second stage software component from one another to identify a faulty stage (fig. 1 item #256 and fig. 2).

Therefore, it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to incorporate the teaching of Hirosawa as stated above with the Ullman in order to store the detected faults with the assigned identifiers and fault descriptions.

One of ordinary skilled in the art would have been motivated because the use of identifiers is well known and that is to identify and distinguish the events from one another.

As per claim 20, Ullman in view of Sato does not disclose the process of associating a fault description with each of the stage identifiers for transmission to a management system via a communications network. Hirosawa from the same field of endeavor discloses the process of associating a fault description with each of the fault and identifier for transmission to a management system via a communications network (fig. 6 and fig. 8, col. 17 L5-32). Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to incorporate the teaching of Hirosawa with Ullman in view of Sato, in order to associate

the fault descriptions with each of the stage identifiers. One of ordinary skilled in the art would have been motivated because it would have provided an administrator a detailed description of the fault, so that an appropriate action could be taken based on information provided.

Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ahmed (U. S. Patent No. 6,813,634 B1) in view of Neimat et al., (hereinafter Neimat, U. S. Patent No. 6,012,059).

As per claim 23, Ahmed does not explicitly disclose the process of storing the status code from an output of the remote software module as a dummy transaction in the database and retrieving the status code as the dummy transaction in the database and feeding the retrieved status code for transmission to the base data processing system if the database provides a logical data path of continuity for the status code.

Neimat discloses the process of storing the data message as a dummy transaction in the database and retrieving the data message as the dummy transaction in the database and transmitting the data message to the replicated site (fig. 4 item #56, fig. 6 item #72; col. 4 L13-21 and col. 5 L63-67; col. 4 L30-60: note that if the message can be stored, retrieved and transmitted from a database, than the inherently provides a logical data path of continuity for the status code).

Therefore, it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ahmed in view of Neimat, in order store the data message or status code as a dummy transaction in the database and feeding the status code for transmission to the base to check if the database provides a logical data path continuity for the status code.

One of ordinary skilled in the art would have been motivated because it would have enabled the synchronization of the events in the system (Neimat, col. 4L13-17). It would have also provided an error detecting means and checking the operating status of the database.

(10) Response to Argument

The examiner summarizes various arguments presented by the appellant and addresses replies individually.

In an appeal brief, appellant argues in substance that:

a. Appellants' specification adequately enables the invention of claim 1 to one of ordinary skill and reasonably conveys to one of ordinary skilled in the art that Appellants had possession of the invention of claim 1 (see brief, page 16-17, paragraph VII.A.1).

In response to argument [a], Examiner disagrees.

Appellant stated on page 16-17 of brief, "step s54 of fig. 8 and page 30, lines 17-23 indicates that in one embodiment the fault detector 165 detects data message or a derivative thereof to determine the flow of the data between logical nodes...thus the fault detector "inherently" determines whether the data message or its derivative flows through at least one of the first stage software component...if the fault detector determines that a data message is present at an earlier stage and absent at a later stage, the fault detector may determine that...by detecting the existence of the data message at an input of a software stage, the fault detector is able to determine if the data message flows entirely through the same software stage by whether or not it detects the data message..."

The appellants' assertion that "by detecting the existence of the data message at an input of a software stage, the fault detector is able to determine if the data message flows entirely

through the same software stage" is certainly not persuasive because the invention is based on detecting whether the flow of the data message or a derivative thereof is blocked or disrupted (see applicant's summary).

Furthermore, appellant has relied on fig. 8 step S54 and fig. 6 to provide the support for detecting whether the data message or a derivative thereof flows entirely through at least one of the first stage software component and the second stage software component, however step s54 simply states, "detecting the data message or a derivative at a group of logical nodes within the installed remote software module to determine flow of the data message, or a derivative thereof, between the logical nodes and, hence, flow through at least one of the first stage software component and the second stage software component". There is simply no indication of the fact wherein the data message flows entirely through the first and second stage software components.

b. There is no mention in Ahmed that the reply is the same ping that was originally sent (see brief, page 19).

In response to argument [b], Examiner disagrees in light of the following reasons:

First, the rejected claims does not indicate that the data message or status code (as per claim 22) outputted from an output of the remote software module is the same data message or status code (as per claim 22) that was inputted into the remote software module.

The claim 22 states:

A method of monitoring a business-to-business system, the method comprising:
transmitting a status code from a base data processing system to a remote data processing system via a communications network;
receiving the status code at a data receiver in the remote data processing system inputting the status code into a remote software module of the remote data processing system;
determining whether the remote software module provides a logical data path of continuity to the status code;

outputting the status code from an output of the remote software module if the determining determines that the remote software module provides a logical data path of continuity to the status code; and

transmitting the outputted status code back to the base data processing system via the communications network as feedback indicative of the proper end-to-end continuity of communications in a business-to-business environment.

There is simply no mention of the fact that the outputted status code is the same status code that was inputted in the remote software module.

Technically, "software" is a computer program that is configured to conduct a process, which takes a value and/or parameter as an input, processes the value and/or parameter, and outputs the results based on the input.

If the software operates properly, the software will take an input, process the input and output a result based on the inputted value. In a case where the software has a fault or an error, the software will not be able to produce a result or at least will disrupt the results.

In the instant claim, the remote software module takes the status code as an input, and outputs the results in form of another status code, which is then transmitted back to the base data processing system as a feedback, in a similar way as disclosed by Ahmed's ping and reply fault detection technique. In Ahmed, the software module receives the ping message, processes the ping message, and outputs the reply message.

Secondly, the specification fails to teach or suggest that the outputted status code is the same status code as the inputted status code (see applicant specification, page 23 lines 2-17, page 24, lines 4-27, page 26, lines 17-29, page 28 lines 14-18).

Accordingly, claim 22 is clearly anticipated by Ahmed.

c. The rejection of claim 22 is improper for the additional reason that it is not relying on a single embodiment to anticipate the claims (see brief, page 20).

In response to argument [c], Examiner disagrees because applicant is misinterpreting the recited portion of the prior art.

Fig. 2 and fig. 3 of Ahmed is the same system, figure 3 basically shows the further steps involved in the invention.

Therefore for the at least this reason, the anticipation by Ahmed is proper.

d. To show inherency, extrinsic evidence must make it clear the missing matter is necessarily present in the reference and it would be so recognized by one of ordinary skilled in the art (see brief, page 21).

In response to argument [d], Examiner agrees with the appellants assertion that the extrinsic evidence must make it clear the missing matter is necessarily present in the reference and it would be so recognized by one skilled in the art.

Applicant specification states, "each remote software module may be organized into multiple stages...for example, the software module 160 may include the first stage software component 161, the second stage software component 162, and the third stage software component 163...Each software component (e.g. 161, 162, or 163) may represent a different program, subroutine, program module, or some other divisible portion of the remote software module (see applicant specification, page 28 lines 5-25).

Examiner in the rejection clearly stated that the PCs are inherently associated with the software modules such as "operating system" that usually includes plurality of software stages cascaded with each other (page 7).

As extrinsic evidence, please consider the following:

First, without the operating system, the PC would not be able to operate and it's fairly known in the art that in order for a PC to operate, it would require an operating system, i.e. a software module.

Secondly, the operating system is a software that controls the allocation and usage of hardware resources such as memory, CPU time, disk space and peripheral devices. The software includes subroutines and/or programs integrated together for controlling different parts of a computer system such as memory and peripheral devices. Without these subroutines and programs, the operating system would not be fully functional; hence the PC would not operate.

Third, all of the above facts are fairly recognized by one of ordinary skilled in the relevant art.

Also, note that the appellant has not provided any evidence in disproving the examiner's reasoning of inherency.

Even though Ahmed inherently taught the remote software module including plurality of software routines or stages, examiner had not relied on the fact and had introduced Sato in order to disclose such a limitation (see the rejection).

e. Ahmed fails to disclose "a fault detector associated with the first software stage component and the second software stage component" that detects a "a fault in the remote software module by detecting whether the data message or a derivative thereof flows entirely through at least one of the first stage software component and the second stage software component" as recited in claim 1. There is no disclosure in Ahmed that the ping message flows through entirely through at least one of the first stage software component and the second stage software component (see brief, page 22).

In response to argument [e], Examiner disagrees because of the following reasons:

The appellant stated, "the office action has asserted at page 7 that the ping message flows through the software module in the PC and the reply is sent. The question is not whether the ping message flows through the software module, but whether the ping message flows "entirely" through at least one of the first stage software component and the second stage software component. There is no disclosure in Ahmed that the ping message flows entirely through a stage software component".

Based on the above paragraph, Examiner takes this assertion to mean that the appellant is admitting that the ping message flows through at least one of the first stage software module and the second stage software module, however it does not Ahmed does not flow "entirely" through the software stages.

First the specification fails to provide any sufficient support for the claimed and argued limitation (see 35 USC 112, first paragraph).

Furthermore, appellant stated on page 16-17 of brief, "step s54 of fig. 8 and page 30, lines 17-23 indicates that in one embodiment the fault detector 165 detects data message or a

derivative thereof to determine the flow of the data between logical nodes...thus the fault detector "inherently" determines whether the data message or its derivative flows through at least one of the first stage software component...if the fault detector determines that a data message is present at an earlier stage and absent at a later stage, the fault detector may determine that...by detecting the existence of the data message at an input of a software stage, the fault detector is able to determine if the data message flows entirely through the same software stage by whether or not it detects the data message..."

The appellants' assertion of "detecting the existence of the data message at an input of a software stage, the fault detector is able to determine if the data message flows entirely through the software stages", is certainly not persuasive because the invention is based on detecting whether the flow of the data message or a derivative thereof is blocked or disrupted (see applicant's summary).

On the other hand, Ahmed discloses the process of sending a ping message to a software module, wherein the software module processes the ping message entirely, and outputs a reply in response to the ping message if the software module is operational (see fig. 1-2).

As such, Ahmed does teach the process wherein the ping message flows entirely through the software component(s).

f. The rejection is improper for the additional reason that there is no motivation in either Ahmed or Sato to alter Ahmed (see brief, page 28-31, paragraph (b-e)).

In response to argument [f], examiner disagrees because examiner has clearly stated the motivation for every combination in the rejection (see the grounds of rejection).

g. Sato does not cure deficiencies of Ullman in that Sato does not disclose nor suggest altering Ullman so that Ullman detects "the data message or a derivative at a group of logical nodes within the installed remote software module to determine flow of the data message, or a derivative thereof (see brief, page 32-33, paragraph 2).

In response to argument [g], Examiner disagrees.

Sato does suggest altering Ullman so that Ullman detects the data message at a group of logical nodes within the installed software module to determine the flow of data message.

Sato expressly teaches the process of detecting the data message at a second operating system, i.e. detecting the existence of the data message between the logical nodes (see fig. 8).

Therefore it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Ullman in order to detect the data messages at a group of logical nodes within the installed software module since Sato teaches the process of detecting the data message between the logical nodes.

And, one of ordinary skilled in the art would have been motivated because it would have monitored the software faults within the monitored computer (see Sato, col. 8 L28-52).

h. Short et al., does not suggest altering Ullman either (see brief, page 33-34, paragraph 3).

In response to argument [h], Examiner disagrees because Short expressly teaches the process of sending the periodic messages, called heartbeats (as in applicant's invention, see specification, pages 24-25) to resources in order to provide a mechanism for detecting that the communication path is good and operational (see Short, col. 5 L1-10).

Short explicitly states the motivation as detecting whether the software module is operational or not (col. 5 L1-5).

Therefore, Short does suggest altering Ullman as set forth above.

(11) Related Proceeding(s) Appendix

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Kamal Divecha
Art Unit 2151
September 18, 2006.

Conferees:

William Vaughn

~~Jason Cardone~~

John Follansbee
SPE 2154

WILLIAM VAUGHN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100